AFFDL-TR-78-18
Volume I

MONTE CARLO BAYESIAN SYSTEM RELIABILITY — AND MTBF-CONFIDENCE ASSESSMENT, II

Volume I: Theory

MITCHELL O. LOCKS
OKLAHOMA STATE UNIVERSITY
STILLWATER, OKLAHOMA 74074

MARCH 1978

TECHNICAL REPORT AFFDL-TR-78-18, Volume I
Final Report March 1976 — December 1977

AIR FORCE FLIGHT DYNAMICS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

H. Leon Harter

H. LEON HARTER
Project Engineer

Richard D. Krobusek

RICHARD D. KROBUSEK, Major, USAF
Chief, Analysis & Optimization Branch

FOR THE COMMANDER

Holland B. Lowndes, Jr.

HOLLAND B. LOWNDES, JR.
Acting Chief, Structural Mechanics Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFFDL/FBRD , W-PAFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

AIR FORCE/56780/6 June 1978 — 450

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFFDL-TR-78-18, Volume I	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment, II	5. TYPE OF REPORT & PERIOD COVERED: Final Report, March 1976 - December 1977
6. PERFORMING ORG. REPORT NUMBER	
7. AUTHOR(s): Mitchell O. Locks, assisted by Keun K. Lee	8. CONTRACT OR GRANT NUMBER(s): F33615-76-C-3094
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Oklahoma State University, College of Business Administration, Stillwater, OK 74074	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F, 2304N104
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Flight Dynamics Laboratory/FBRD, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433	12. REPORT DATE: March 1978 13. NUMBER OF PAGES: 59
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

This report is a sequel to AFFDL-TR-75-144, "Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment," 1975, and a companion to AFFDL-TR-78-18, Volume II, "SPARCS-2 Users Manual".

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Reliability	Poincaré's theorem	Modularization	Minimal paths
Monte Carlo	Inclusion-exclusion	Poisson process	Minimal cuts
Assessment	Bernoulli process	Beta distribution	Probability
System assessment	Conjugate prior distribution	Negative-log gamma distribution	Simulation
Complex systems			

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

SPARCS-2 (Simulation Program for Assessing the Reliabilities of Complex Systems, Version 2) is a PL/1 computer program for assessing (establishing interval estimates for) the reliability and the MTBF of a large and complex system of any modular configuration. The system can consist of a complex logical assembly of independently failing attribute (binomial-Bernoulli) and time-to-failure (Poisson-exponential) components, without any regard to their placement. Alternatively, it can be a configuration of independently failing modules, where each

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Block 19 (continued)

Logic	MTBF assessments
Subsystems	Pass-fail components
Failure modes	Time-to-failure components
Component assessment	

Block 20 (continued)

module has either or both attribute and time-to-failure components. The raw data for assessments are the component failure history data and the system configuration. The historical data are "successes and failures" for binomial-Bernoulli components and "failures and testing time (normalized to 'mission equivalent units')" for time-to-failure components. The configuration data consist of a list or lists of minimal paths ("minimal path sets" or "tie sets"), or else a list of minimal cuts ("minimal cut sets"), for the system as a list of modules, and for each module as a list of components. If the MTBF assessment option is selected, the system "mission time" is also needed. The underlying mathematical model, identical with that incorporated into the first version of the SPARCS program described in [5], is an amalgamation of Boolean logic, probability, and Bayesian and Monte Carlo techniques. The system reliability, a numerical-valued function of the component reliabilities, is derived by the method of inclusion-exclusion (IE), also known as Poincaré's theorem, from the minimal paths or the minimal cuts. The failure-history data are "sufficient statistics," for the parameters of Bayesian conjugate prior distributions (c.p.d.'s) on the component reliabilities, "beta" for attributes and "negative-log gamma" for time to failure.

PREFACE

This report of work performed under Contract F33615-76-C-3094 has two divisions. Since each of these divisions deals with a separate aspect of the work performed under the contract, they are presented herein as two separate papers, with independent lists of references. The introduction presents the historical background for this project, including prior work directly related to software development, input data documentation, and an example. Division 1 is a broad-range tutorial on system reliability analysis, covering a very broad range of related subject matter. Division 2 is specialized to a discussion of the beta and gamma random deviate generators incorporated into the software developed under this project and documented in the companion "SPARCS-2 Users Manual," AFFDL-TR-78-18, Volume II.

This work was performed under work unit 2304N104, System Reliability - Confidence Assessment, with Dr. H. Leon Harter as project engineer.

TABLE OF CONTENTS

INTRODUCTION

SPARCS-2 (Simulation Program for Assessing the Reliabilities of Complex Systems, Version 2) is a PL/1 computer program for assessing (establishing interval estimates for) the reliability and the MTBF of a large and complex system of any modular configuration. The system can consist of a complex logical assembly of independently failing attribute (binomial-Bernoulli) and time-to-failure (Poisson-exponential) components, without any regard to their placement. Alternatively, it can be a configuration of independently failing modules, where each module has either or both attribute and time-to-failure components.

The raw data for assessments are the component failure history data and the system configuration. The historical data are "successes and failures" for binomial-Bernoulli components and "failures and testing time (normalized to 'mission equivalent units')" for time-to-failure components. The configuration data consist of a list or lists of minimal paths ("minimal path sets" or "tie sets"), or else a list of minimal cuts ("minimal cut sets"), for the system as a list of modules, and for each module as a list of components. If the MTBF assessment option is selected, the system "mission time" is also needed.

The underlying mathematical model, identical with that incorporated into the first version of the SPARCS program described in [5], is an amalgamation of Boolean logic, probability, and Bayesian and Monte Carlo techniques. The system reliability, a numerical-valued function of the component reliabilities, is derived by the method of inclusion-exclusion

(IE), also known as Poincaré's theorem, from the minimal paths or the minimal cuts. The failure-history data are "sufficient statistics," for the parameters of Bayesian conjugate prior distributions (c.p.d.'s) on the component reliabilities, "beta" for attributes and "negative-log gamma" for time to failure.

SPARCS assesses by Monte Carlo. At each trial, for each component, a value of the reliability is generated from the c.p.d. and substituted into the system function, to obtain a value of the system reliability for that trial. The resulting "empirical" distribution of system reliabilities, obtained over a series of trials, provides the basis for an assessment. Percentage points on that distribution are interpreted as system reliability confidence limits. The corresponding MTBF confidence limits are calculated, based on the simple relationship between the reliability and the MTBF.

SPARCS was developed at Oklahoma State University in a series of stages. Initially J. L. Burris [1] prepared an estimation program MAPS (Method for the Analysis of the Probabilities of Systems) to calculate the system reliability as an exact function of the component reliabilities, based upon system logical input data. MAPS was modelled to a great extent after a similar system programmed in FORTRAN and assembler language called SCOPE (System for Computing Operational Probability Equations) [2, 3], which had been developed at Rockwell International in the 1960s under a NASA contract. Certain significant improvements were incorporated into MAPS, as follows:

1. PL/1 programming: this made it possible to develop a fully "portable" program package with efficient binary digit manipulation capability, in a more adaptable higher level language than FORTRAN.

2. Modularity: this made it possible to generate the equation and calculate the probability for a system of substantially larger sizes than can be processed with SCOPE.

3. "Minimal path" or else "minimal cut" calculations: by taking advantage of the dual relationship between the paths and the cuts, the same software can be used either to obtain a system reliability function of the component reliabilities from the minimal paths, or else an unreliability function of the component unreliabilities from the minimal cuts.

Under Contract F33615-74-C-4072, Cooley [4] programmed the initial version of the SPARCS system for system reliability- and MTBF-confidence assessment. The same logical configuration data are needed as for MAPS. Instead of inputting the component probabilities as in MAPS, however, the component failure-history data are used, for both attribute and time-to-failure components. The attribute data are accumulated successes and failures; the time-to-failure data are accumulated failures and testing time.

The initial version of SPARCS, as documented in [4] and [5], in our opinion represents a landmark and a significant state-of-the-art improvement over other programs designed for estimating or assessing system reliability. The results of an assessment can be verified in the sense that they are reproducible; that is, one can input either the minimal paths or else the minimal cuts and obtain a reasonably similar assessment for any given set of input data. In addition, accurate reproducible assessments can be obtained with many fewer trials (20-100) than by older Monte Carlo programs, which sample missions rather than reliabilities and therefore require many thousands of trials.

There are some aspects of the initial version of SPARCS which made it relatively difficult to use. The most significant problem in this

regard was the fact that in order to generate beta and gamma deviates, it was necessary to set up calls to proprietary IMSL routines programmed in FORTRAN and assembler languages. This is a clear interference with the portability feature, since the rest of the program is PL/1 and nonproprietary. It also made the JCL language input more complicated. In addition, the program was also overly long, and required too much core space and running time.

Oklahoma State University was awarded a second contract, F33615-76-C-3094, to make improvements in SPARCS, as well as to perform certain related studies, such as error analysis of the random deviate generators and model validation. In the process, we developed a program, now called SPARCS-2, which not only has corrected major deficiencies in the original version, but is also more efficient and much more serviceable from the viewpoint of the user.

The following represent the improvements in SPARCS-2:

1. The new beta and gamma generators, CABTA and RGAMA respectively, both employing a "rejection" technique, are much faster than the IMSL routines in the original version. These generators are described in greater detail in Division 2 of this report.

2. The program is more compact, about ½ the original size, and makes extensive use of the dynamic storage allocation feature of the PL/1 language.

3. Core requirements are less than in the original version and are variable; for example, very small problems can take about 100K.

4. There is an improved "super modularity" capability. Modules with minimal-cut unreliability calculations can be mixed with those having minimal-path reliability calculations. However, all output has been standardized to system reliability or "probability of success," regardless of the form in which the input data are presented, and whatever the configuration of modules or elements within modules.

5. Larger systems can be handled, and limitations are more clearly spelled out. A system can consist of a configuration of up to 128 modules or components, with up to 256 minimal states. Likewise, a module within the system can have 128 components, and 256 minimal states. A probability equation can have up to 3500 terms.

6. Internal documentation is very much improved, based on "structured programming" concepts, to facilitate the preparation of input data, and for the benefit of users who may have to make alterations in capacity or in some of the internal procedures.

REFERENCES

[1] Burris, Jimmy L., "Model for the Analysis of the Probabilities of Systems," MBA research report, Department of Administrative Sciences, Oklahoma State University, 1972.

[2] North American Rockwell Corporation Space Division, "Exact Minimal-Path Techniques for Determining System Reliability," Program MFS-16499; available through NASA's COSMIC, University of Georgia, Athens, GA 30601.

[3] North American Rockwell Corporation Space Division, "System for Computing Operational Probability Equations (SCOPE): Version II," Program FMS-24035; available through NASA's COSMIC, University of Georgia, Athens, GA 30601.

[4] Cooley, John W., "Simulation Program for Assessing the Reliability of Complex Systems (SPARCS)," Ph.D. dissertation, Oklahoma State University, April, 1976.

[5] Locks, Mitchell O., "Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment," Air Force Flight Dynamics Laboratory, Wright-Patterson AFB, Ohio, AFFDL-TR-75-144.

[6] Brown, J. R. and M. Lipow, "Testing for Software Reliability," TRW Systems Engineering and Integration Division, One Space Park, Redondo Beach, CA 90278, TRW-SS-75-02, January, 1975.

DIVISION 1

SYSTEM RELIABILITY ANALYSIS: A TUTORIAL

by

Mitchell O. Locks
Professor of Management Science
Oklahoma State University

ABSTRACT

This paper deals with the logical formulation of a system for purposes of reliability analyses and both exact and approximate methods of calculating the system reliability. The first part deals with the logical concepts, and the second part with probability calculations.

The logical formulation in Part 1 starts from first principles. The universal set $\mathcal{U}$ of system states is a "Boolean algebra". The "power set" $\mathcal{P}$ is the set of subsets of $\mathcal{U}$. The subset of system success states or "paths" is a "lattice" within $\mathcal{U}$, also an element of $\mathcal{P}$, represented by a Boolean polynomial. The terms of the polynomial are monomials which give the indicators for the "sublattices" or "complete subsets" of $\mathcal{U}$ within the lattice of paths. The optimal representation of this lattice polynomial is in the minimalized form; the terms of the minimalized lattice of paths are called the "minimal paths". Similarly, the subset of system failure states or "cuts" is a lattice polynomial whose terms are sublattices within the lattice of cuts; the terms of the minimalized lattice polynomial are the "minimal cuts". Logically consistent systems (also known as "coherent structures") have certain properties with respect to the partial ordering of paths and cuts. The concepts of Boolean logic, minimalization, etc., apply to systems that do not have the "consistency" property, as well as to those that do have it.

The second part of this paper deals with ways of using a minimalized lattice polynomial to derive a numerical-valued probability function from which to calculate the system reliability: also computer software, error bounds and approximations. The reliability is the probability that the actual state of the

system is an element of the lattice of paths. The exact probability is derived from the minimalized lattice polynomial of paths by the method of inclusion-exclusion, also known as Poincaré's Theorem. Dually the probability of failure, or system unreliability, is derived from the minimalized lattice of cuts. Modularization and/or inversion can be helpful in keeping the probability function reasonably small in size. Computer software is available both to generate the probability polynomial and to calculate either the reliability or unreliability. Approximations can be used based upon simplifying assumptions which delete low probability terms from the function. Conservative error bounds are obtainable for some models. The approximation techniques include serializing methods ("single-point failures" and "parts count"), very large system approximations for fault-tree applications and the Esary-Proschan bounds.

INTRODUCTION

System reliability analysis is closely related to Boolean logic. From George Boole's classic 1854 book, The Laws of Thought [9], it can be seen that the kind of problem Boole was trying to solve

> ... to make that method itself the basis of a general method for the application of the mathematical doctrine of Probabilities ... [9, p. 1]
>
> ... it is always possible ... to express the event whose probability is sought as a logical function of the events whose probabilities are given ... [9, p. 15]

is essentially the same problem that a system reliability analyst tries to solve. In this paper, the objective is logically symbolizing the success or failure of a system as a function of success or failure for various combinations of its components, and evaluating the probability of success, that is, the system reliability as a function of the component reliabilities.

Boole made some errors. His explanation of "inversion" involved a clumsy use of series expansions and divisions by logical zero, in contrast to the more elegant De Morgan's theorems. Also, he did not exhibit a deep understanding of "duality". Subsequent work by others, based on Boole's foundations, led to many significant developments in mathematics, symbolic logic and philosophy, including set theory, particularly lattice theory and the theory of partially ordered sets, Boolean algebra, and minimalization by Quine [46, 47, 48]. The contributions of Boolean logic to applied disciplines such as computer science and electrical engineering, particularly through the pioneering work of Shannon [54], are well known and recognized.

Logically based system reliability analysis appears to have been initiated by von Neumann in connection with his search for ways of explaining

how large digital computers having many thousands of unreliable components such as vacuum tubes could operate reliably. In a paper published in 1956 based on lectures given in 1952 at California Institute of Technology, von Neumann [59] showed that by the use of redundancy, it is possible to build and maintain a complex system having a greater reliability than any of its components.

Moore and Shannon [41] used essentially parallel-series circuits consisting exclusively of idealized identical relays all with identical reliabilities to develop functions relating the reliability of a system R to that of the components. Bounds on this relationship were developed showing at what point R could be greater than the reliability of the components. Mine [40] generalized this work further by introducing Boolean and set-theoretic notation, and employed linear graph theory to find the functional relationship between system and component reliability. He also developed bounds and conditions under which arbitrarily high R could be obtained.

Birnbaum, Esary, and Saunders [8] extended the Moore-Shannon approach to the general class of "coherent" systems that have certain logical consistency properties. Their paper introduced terminology which has been widely used, such as "minimal paths", "minimal cuts", and essential (relevant) components". Esary and Proschan [16, 17] obtained approximations for the reliability of coherent systems, including upper bounds based on the minimal paths and lower bounds based on the minimal cuts. The theory of coherent systems based on this general approach is discussed by Barlow and Proschan [4, pp. 1-51]. A review of the literature, with special

reference to the contributions of Z. W. Birnbaum and his students and colleagues, is given by Saunders [52].

The research reported upon in the body of this paper deals primarily with generalized approaches that apply not only for a complex system of any configuration, but also if every component is different. Several large scale computer programs of this type were prepared in the early 1960s in connection with the U. S. space program. Some of these programs employed Monte Carlo approaches. As a general rule, very little documentation is available as to what theory was employed or what was done.

A FORTRAN software package with the acronym SCOPE (System for Computing Operational Probability Equations) was developed about 1965 at the Rockwell International Corporation. SCOPE provides an exact system reliability function of the component reliabilities for a complex system of any configuration but of limited size, based on either the minimal paths or else the minimal cuts. This function is derived by the method of inclusion-exclusion, also known as Poincaré's theorem. To obtain the system value, simply substitute the component values into this function. The SCOPE software is available through NASA [44], and the mathematical theory is given in [28], including some comparisons with the Esary-Proschan bounds.

An improved version of SCOPE was prepared in 1972 by Burris [11] at Oklahoma State University with the acronym MAPS (Method for the Analysis of the Probabilities of Systems). Instead of FORTRAN, it is programmed in PL/1, which is better adapted to binary-digit manipulation. It also incorporates a modularity feature so that the system can optionally be processed as a complex configuration of independent modules, each module consisting of a complex configuration of independently failing elements. Consequently,

there is simultaneously both an increase in capacity and a saving in computer time, over the requirements of the parent SCOPE program.

A further extension of MAPS is SPARCS (Simulation Program for Assessing the Reliabilities of Complex Systems) programmed by Cooley [13] at Oklahoma State University under Air Force Contract F33615-74-C-4072. This uses Monte Carlo combined with Bayesian techniques to assess (provide a schedule of confidence levels for all values of R between 0 and 1) both R and the MTBF (mean time between failures) of a complex system of any configuration consisting exclusively of pass/fail and/or time-to-failure components, or any mixture of these, from failure-history data. Both MAPS and SPARCS are described in [29]. A more efficient version of SPARCS, called SPARCS-2, has been prepared by Lee [26].

A related and parallel effort to SCOPE and its daughter programs (or SPARCS and its parents) is the development of the fault-tree methodology. Initially designed for aerospace applications at Bell Labs [6] in 1961, and subsequently also at Boeing Airplane Company, as an aid to engineers in analyzing sequences of events leading to system failure, it has recently been used extensively for reactor safety studies by the Atomic Energy Commission and its successor, the Nuclear Regulatory Commission. A recent but historic document reporting the results of extensive applications of the methodology is the "Rasmussen Report WASH-1400" [55], particularly Appendix II, "Fault-Trees". Descriptions of the methodology are given by Mearns [37], Haasl [23], Eagle [14], Schroder [53], Barlow and Lambert [3], Vesely [56, 57, 58], and Barlow and Proschan [4, pp. 264-266].

A fault tree is a Boolean-equivalent diagrammatic representation of all the ways of failing a complex system through combinations of failures

and repairs of one or more components. It is derived from block diagrams, schematics, and/or blueprints, and logical analysis of the interrelationships of the elements. "Minimal cut sets" are obtained from the tree. By substituting the component values, an estimate of R, the system availability or the failure rate are obtained. A rather substantial library of fault-tree methodology computer programs is available, including programs which build the trees, find the cut sets, or perform numerical evaluations. Vesely made some major contributions with the development of the PREP and KITT codes [56, 57]. Salem, Apostolakis, and Okrent [51, pp. 34-45] and Worrell and Burdick [62] give reviews of the available software.

In general, the fault-tree methodology incorporates the same Boolean and probabilistic theory that SPARCS and its parent programs do. However, because the methodology is generally applied to large systems having only low failure-rate components, various types of "rare event approximations" are employed, to save computer time. These include both Monte Carlo and deterministic selections of components and cut sets according to importance, deleting higher order intersection terms in the probability equation, and basing calculations on failure rates rather than probabilities. "Importance" measures are discussed by Nagel [42], Nagel and Schroder [43], Lambert [25], and Mazumbar [36], and Barlow and Proschan [4, pp. 26-29].

This article is a review of the state of the art of evaluating R for a complex system as a function of the reliabilities or failure rates of the elements. It develops the common Boolean theoretical structure which underlies all the different methodologies, and shows some of the similarities of and the differences between exact methods and approximations. There are two parts: Part 1, on "Logical Formulation" and Part 2 on "Probability Calculations".

Part 1 of the paper gives the logical formulation, starting from set-theoretic first principles, from the viewpoint of lattice theory. We describe the universal set $\mathfrak{U}$ of system states and the power set $\mathcal{P}$ of lattices or events which are subsets of $\mathfrak{U}$. Events such as system success or system failure are collections of success states, called "paths", or else failure states, called "cuts", described by lattice Boolean polynomials. The terms of the minimalized form of the "success" polynomial are the "minimal paths" and for the "failure" polynomial the "minimal cuts" where each term denotes a sublattice.

Part 2 describes how to calculate the system reliability. First a probability function, a numerical-valued function of the component probabilities, is derived from the lattice polynomial by the method of inclusion-exclusion. Then the component probabilities are substituted into this function. Part 2 discusses exact methods, error bounds, approximations, serializing methods and the fault-tree methodology.

Part 1: LOGICAL FORMULATION

BOOLEAN ALGEBRA

Components, System States, and the Universal Set

A system has n zero-one (binary-valued) components, each representing in reliability terms a potential failure mode or type. The value "0" denotes a failure condition and "1" a success. The binary representation is not meant to include only attribute failure modes. Both time-to-failure components and those for which success or failure is defined by exceeding or not exceeding a critical value of a variable can be included, since these too can either be failing or operating successfully.

A state of the system is a binary n-vector, with each two-valued component denoting a value for the corresponding failure type. Since there are n components and two choices for each component, there are 2^n different n-vectors, each representing a "unit event" or "simple event" which is a possible state of the system. The collection of these vectors is called the universal set $\mathfrak{U}$. Those n-vectors which represent system "success" are called "paths" and those for "failure" are called "cuts".

The Boolean operations of addition, multiplication, and inversion are performed on the n-vector elements of $\mathfrak{U}$, component by component, in the usual way: $0 + 0 = 0$; $0 + 1 = 1 + 1 = 1$; $0 \cdot 0 = 0 \cdot 1 = 0$; $1 \cdot 1 = 1$; $\bar{1} = 0$, $\bar{0} = 1$. The Boolean sum of any two or more n-vectors is their least upper bound l.u.b., and the Boolean product is the greatest lower bound g.l.b.

Let $X = (x_1, \ldots, x_n)$, $Y = (y_1, \ldots, y_n)$, $Z = (z_1, \ldots, z_n)$, x_i, y_i, $z_i = 0, 1$, $i = 1, \ldots n$, be any three binary n-vector elements of $\mathfrak{U}$. Both the Boolean operations of addition and multiplication are associative ($(X + Y) + Z = X + (Y + Z)$, $(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$) and commutative ($X + Y = Y + X$, $X \cdot Y = Y \cdot X$) and the system is distributive with respect to the operations of addition and multiplication ($(X + Y) \cdot Z = XZ + YZ$). Since there is a unique "zero" element $(0, \ldots, 0)$, all components failed, a unique "one" element $(1, \ldots, 1)$, all components working, and all Boolean operations are performed on any two or more n-vector elements of $\mathfrak{U}$ to yield another n-vector, the universal set $\mathfrak{U}$ is a Boolean algebra (see Halmos [24, p. 5, 40]).

Comparison and Partial Ordering

The universal set $\mathfrak{U}$ is partially ordered, with the $\leq$ operation. For any pair of n-vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$, $X \leq Y$ (X is smaller than or equal to Y) if $X \cdot Y = X$ ($x_i = 1 \Rightarrow y_i = 1$). This also implies that $X + Y = Y$. Equality, strict inclusion, and strict noninclusion are defined as follows, respectively:

$$X = Y: \quad X \leq Y, \; Y \leq X$$

$$X < Y: \quad X \leq Y, \; Y \neq X$$

$$X \not< Y: \quad X \cdot Y < X.$$

Subsets and Lattices

A lattice L is a nonempty (meaning: it contains at least one element) subset of $\mathfrak{U}$, for which every pair of elements (i.e., binary n-vectors)

has both a g.l.b. Boolean product and l.u.b. Boolean sum (see Birkhoff [7, p. 6]). In the probability sense, a lattice is also known as an "event". Since every pair of n-vectors has both a g.l.b. and l.u.b., L also has a g.l.b. Boolean product and l.u.b. Boolean sum for all the n-vectors in the lattice. Both the g.l.b. and l.u.b. are members of $\mathcal{U}$, but not necessarily of L.

The differences between a lattice and a Boolean algebra have to do with both complementation and membership. In a lattice, the only operations are Boolean addition and multiplication; in a Boolean algebra, there is also complementation or inversion. With respect to membership, the n-vector resulting from an operation or sequence of operations upon any two or more members of the same Boolean algebra is in the algebra. By contrast, for a lattice L, the result of either operation, addition or multiplication, on any two or more members of L does not necessarily result in a member of L.

The Power Set

The set of lattices generated by forming subsets of $\mathcal{U}$ is called the "power set" $\mathcal{P}$. For a system with n components, $\mathcal{P}$ has 2^{2^n} lattices. This can be rationalized in essentially the same way we explain why $\mathcal{U}$ has 2^n elements. $\mathcal{P}$ has a set of subsets created by forming collections of n-vectors. For every lattice in $\mathcal{P}$ and every n-vector in $\mathcal{U}$, there are two choices: either the n-vector is in the lattice, or else it is not. Since these two choices are available for every one of the 2^n n-vectors in $\mathcal{U}$, there are 2^{2^n} lattices in $\mathcal{P}$.

Complete Subsets: Sublattices

The most convenient way of representing subsets of $\mathcal{U}$ is as inclusive-or unions of "sublattices" or "complete subsets". According to Rutherford [50, p. 9], a sublattice S is a lattice which includes both the g.l.b. and l.u.b. for every pair of elements in S. Let $X \in S$, $Y \in S$ be any n-vectors; then $X + Y \in S$, $X \cdot Y \in S$. Equivalently, a sublattice is a lattice which contains both its own g.l.b. and l.u.b. and every n-vector element in the interval between. Every sublattice is characterized by m fixed-valued components called "indicators", $m \leq n$, which have the same value for every one of the n-vector elements in S. In [34], it is shown that there are 3^n sublattices in $\mathcal{U}$.

Example

A system has two components. The universal set $\mathcal{U}$ has $2^2 = 4$ elements

$$\mathcal{U} = \{00, 01, 10, 11\}.$$

The power set $\mathcal{P}$ has $2^{2^2} = 16$ different subsets of $\mathcal{U}$

$$\mathcal{P} = \{\emptyset, \{00\}, \{01\}, \{10\}, \{11\}, \{00, 01\}, \{00, 10\}, \{00, 11\}, \{01, 10\}, \{01, 11\}, \{01, 10\}, \{00, 01, 10\}, \{00, 01, 11\}, \{00, 10, 11\}, \{01, 10, 11\}, \mathcal{U}\}.$$

As is customary, $\emptyset$ denotes the "empty set" (with nothing in it). There are $3^2 = 9$ sublattices in $\mathcal{P}$

$\{00\}$
$\{01\}$
$\{10\}$
$\{11\}$
$0 \cdot = \{00, 01\}$
$1 \cdot = \{10, 11\}$
$\cdot 0 = \{00, 10\}$
$\cdot 1 = \{01, 11\}$
$\mathcal{U}$.

The Algebra of Sets and Lattices

The power set $\mathcal{P}$ is an algebra of lattices. Algebraic operations are performed on lattices which are elements of $\mathcal{P}$ to form other elements of $\mathcal{P}$. Somewhat comparable to the Boolean operations on n-vector elements of $\mathcal{U}$, these set-algebraic operations are "inclusive-or union ($\cup$, $\vee$, also called 'join')", "intersection ($\cap$, $\wedge$, also called 'meet')" and "complement". Let L_1, L_2, L_3 be any three lattices of $\mathcal{P}$. The union and intersection of operations are associative: $(L_1 \vee L_2) \vee L_3 = L_1 \vee (L_2 \vee L_3)$, $(L_1 \wedge L_2) \wedge L_3 = L_1 \wedge (L_2 \wedge L_3)$; commutative: $L_1 \vee L_2 = L_2 \vee L_1$, $L_1 \wedge L_2 = L_2 \wedge L_1$; and distributive: $(L_1 \vee L_2) \wedge L_3 = (L_1 \wedge L_3) \vee (L_1 \wedge L_2)$.

Let $L_1, \ldots, L_m$ be any m lattices in $\mathcal{P}$, $m \leq 2^{2^n}$. The union $\bigcup_{j=1}^{m} L_j$ consists of all n-vectors in at least one of the lattices $L_1, \ldots, L_m$. The intersection $\bigcap_{j=1}^{m} L_j$ consists only of the n-vectors in at least one of the lattices $L_1, \ldots, L_m$. The complement $\bar{L}$ of a lattice L consists of all n-vectors in $\mathcal{U}$ that are not in L.

BOOLEAN POLYNOMIALS AND MINIMALIZATION

A lattice subset of $\mathcal{U}$ which is also an element of $\mathcal{P}$ is characterized by a Boolean polynomial. Each term or monomial, also frequently called "minterm", is a collection of indicators for a sublattice subset of the lattice. The polynomial or form is an "inclusive-or" union of these sublattices. It is customary to employ the logical "or" ($\vee$) symbol to separate terms; sometimes "+" is used instead if there is no ambiguity as to the difference between logic and arithmetic.

For example, a system has three binary-valued components: x_1, x_2, and x_3; x_i denotes that $x_i = 1$ and $\bar{x}_i = 0$. The lattice

$$L = x_1 \vee x_2\bar{x}_3$$

has two sublattices $x_1 = \{x_1x_2x_3,\ x_1\bar{x}_2x_3,\ x_1x_2\bar{x}_3,\ x_1\bar{x}_2\bar{x}_3\}$ and $x_2\bar{x}_3 = \{x_1x_2\bar{x}_3,\ \bar{x}_1x_2\bar{x}_3\}$. This representation is not unique. There are other sublattice configurations, including, for example:

$$x_1\bar{x}_2 \vee x_1x_2 \vee \bar{x}_1x_2\bar{x}_3$$

$$x_1\bar{x}_3 \vee x_1x_3 \vee \bar{x}_1x_2\bar{x}_3.$$

Clearly, because it is shorter and has more coverage per term and the smallest number of indicators in each term, "$x_1 \vee x_2\bar{x}_3$" is better than any of the alternatives.

The "optimal" or "best" representation of a lattice is a "minimal" form, such as "$x_1 \vee x_2\bar{x}_3$" in the above example. This implies the largest coverage in each term, the smallest number of terms, and the smallest average length of a term. In general, minimal forms may or may not be unique. Quine [46, 47, 48] defined the scope of the minimalization problem in terms of finding all the so-called "prime implicants", the terms of all the different alternative minimal forms. In a recent series of papers, Locks [31, 33, 35] introduced the "reversal" method of minimalization, partially based on Quine's methodology, without necessarily finding all the prime implicants.

Coherent and Noncoherent Systems and Minimal States

Birnbaum et al. [8] introduced terminology which has since been widely used, such as "minimal paths", "minimal cuts", etc., in the context of a coherent structure which has certain logical consistency properties appropriate for system reliability analysis. The system has n zero-one elements x_i, $i = 1, \ldots, n$; $x_i = 0$ denotes component failure and $x_i = 1$ success.

The 2^n n-vector elements of $\mathcal{U}$ are divided into two major lattices or "events", the lattice of "paths" L for the success states and its complement, the lattice of "cuts" $\bar{L}$ for the failure states.

For a coherent system, the "zero" element (0, ..., 0), all components failed, is a cut, the "one" element (1, ..., 1), all components successful is a path, and no path can be smaller than a cut (X a path and Y a cut $\Rightarrow$ $X \not< Y$). Otherwise, the system is "noncoherent". In the coherent case, when the lattice polynomial L for the set of success states is in its minimalized form, every sublattice subset of L is represented only by one-valued components and no sublattice is a proper subset of any other sublattice. The terms of the minimalized polynomial for L are called the "minimal paths" or else the "minimal path sets". Similarly, the minimalized polynomial $\bar{L}$ for the lattice of cuts has terms containing only zero-valued components, called the "minimal cuts" or "minimal cut sets".

For example, the minimalized lattice of paths for a two-component parallel case is

$$L = a \vee b$$

and the polynomial of cuts has one term

$$L = \bar{a}\bar{b}.$$

This is an example of coherence. The previous example is noncoherent since

$$L = x_1 \vee x_2\bar{x}_3, \text{ and}$$
$$\bar{L} = \bar{x}_1\bar{x}_2 \vee \bar{x}_1 x_3.$$

This is not coherent because the cut state 011 contains the path 010. While

the literature does not yet explicitly define minimal paths and minimal cuts for noncoherent systems, there would be no conflict with current practices if we called x_1 and $x_2\bar{x}_3$ minimal paths and $\bar{x}_1\bar{x}_2$ and $x_1\bar{x}_3$ minimal cuts.

Minimalization of a coherent system is simpler than for a noncoherent one. The minimal form in the coherent case is unique, and the only simplification operation employed is absorption (example: $abc \vee ab = ab$). By contrast, for a noncoherent system, the minimal form may or may not be unique; also, because complementary components are involved, the process requires the operations of merging (example: $ab \vee a\bar{b} = a$) and redundant indicator elimination (example: $ab\bar{c} \vee ac = ab \vee ac$). Cycling is also needed as described in [31] and [33] to test for both minimality and uniqueness.

The literature on coherent systems and the financial investment that has been made in computer software for computing estimates of the reliability of coherent systems are rather extensive. For further information, refer to the Introduction. Comparatively little theory is available about noncoherent systems. However, there is a computer program by Worrell called SETS (Set Equation Transformation System) for noncoherent minimalization in the context of system reliability fault-tree analysis (see Worrell [61] and Atkinson [1] in [2]). Noncoherence arises whenever there is dependency of one failure mode type upon another. Examples of potential applications of noncoherence include: maintenance, test, repair, and human factors problems, the so-called "common cause" failures [12, 15, 60] and [55, Appendix IV] (failure modes which can simultaneously cause failures of more than one component), and "phased missions" [18] (missions in which success or failure in one phase depends upon success or failure in another phase) (see also Example 3 in [28] and Figure 1 of Powers, Tompkins, and Lapp [45]). Since many of

the same definitions ("minimal path", "minimal cut", etc.) seem to carry over from coherence to noncoherence, and since the common linkage between the two seems to lie in techniques for minimalization, a further development of the theory of noncoherence would be helpful in understanding the properties of both coherent and noncoherent systems.

Part 2: PROBABILITY CALCULATIONS, SOFTWARE, APPROXIMATIONS AND ERROR BOUNDS

INTRODUCTORY

The Boolean foundations of system reliability analysis are presented in Part 1. The transition from logic to probability is the method of inclusion-exclusion (IE). Both exact and approximate IE methods for generalized complex systems are described in Part 2, based on lattice theory and the minimal path-minimal cut approach of Part 1. The universal set U is a probability space, and its lattice subsets are "events". The sublattice events in the minimalized lattice of paths are represented by the "minimal path sets" and those in the minimalized lattice of cuts by the "minimal cut sets". IE generates a system reliability function of the component reliabilities from the minimal paths, or else an unreliability function of the component unreliabilities from the minimal cuts. The component probabilities are substituted into this function to obtain the system probability.

System reliability analysis by exact methods frequently requires excessive computer time. By inverting the logic function or portions of it, or by modularizing the system, in many cases the computer time can be reduced. However, in large scale applications, such as large fault-tree problems or microminiaturized logic circuit "black boxes" consisting of many thousands of elements, cancellations and reductions achieved solely in the context of the exact methodology may still not be enough. Approximations can be employed in those cases if the probability of system failure is small. Part 2 covers exact methods, including computer software, and approximations, including error bounds.

DISCUSSION

The Universal Set as a Probability Space

In order to simplify the calculations, assume that the n components all succeed or fail independently. Under this assumption, the probability of the unit event representing each n-vector element is obtained by multiplying the probabilities of success (reliabilities) of the one-valued components by the probabilities of failure (unreliabilities) of the zero-valued components. Let

$$x_i = 0, 1, \; i = 1, \ldots, n$$

be the logical value of the i-th component; $x_i = 0$ denotes "failure" and $x_i = 1$ denotes "success". Also, let

$$r_i, \; 0 \leq r_i \leq 1, \; i = 1, \ldots, n$$

be the component reliability or probability of success, and $1 - r_i$ the component unreliability or probability of failure. Let

$$X = (x_1, \ldots, x_n)$$

be an n-vector element of the universal set $\mathfrak{U}$. Then

$$\text{pr } X = \prod_{i=1}^{n} r_i^{x_i} (1 - r_i)^{1-x_i}$$

is the probability of the n-vector. It is shown in [28] that under these assumptions $\mathfrak{U}$ is a probability space, since the probability for each element is nonnegative, the sum of the probabilities is one, and the probability of any subset is additive in the probabilities of its mutually exclusive elements.

Sublattice Events

The simplest case is a "sublattice event". The probability of the event is the numerical product of the probabilities of the indicators, those components which have the same value in every n-vector element of the sublattice. Let S be a sublattice with m indicators, x_i, $i = 1, \ldots, m$, $m \leq n$; for each i, r_i is the reliability. Under the independence assumption:

$$\text{pr } S = \prod_{i=1}^{n} r_i^{x_i}(1 - r_i)^{1-x_i}$$

For example, for the sublattice $x_1 x_2 \bar{x}_3 x_5$

$$\text{pr } S = r_1 r_2 (1 - r_3) r_5.$$

Series and Parallel Systems

An important special case of a sublattice event is the so-called logical "series" system. In this case, every component must operate for the system to function successfully. The universal set $\mathcal{U}$ has a single path, the unit event (1, ..., 1), and all other n-vectors are cuts, because the failure of only one component is all that is needed in order for the system to fail. Thus, if there are n components, the system reliability R is the product of the component reliabilities

$$R = r_1 r_2 \cdots r_n.$$

"Series reliability" implies "parallel unreliability" since only one component failure is needed to induce system failure. The dual concept is "parallel reliability"; if only one component works, the system operates successfully. By a similar reasoning, parallel reliability implies "series

unreliability" since the failure of all components is required for system failure and the unreliability $\bar{R} = 1 - R$ is the product of the component unreliabilities $(1 - r_1)\ (1 - r_2)\ \ldots\ (1 - r_n)$.

Compound Sublattice Events

Compound sublattice events are obtained from combinations of unions, intersections, and complements of sublattices and are represented by Boolean polynomials. The most elementary compound event is an "intersection", since the intersection of sublattices is also a sublattice. The intersection L of sublattices $L_1, \ldots, L_m$

$$L = \bigcap_{j=1}^{m} L_j,$$

being a proper subset of all the L_j, is also a sublattice represented by all of the indicators for all of the L_j; in other words, the set of indicators for L is the union of the indicators for the L_j. For example, the intersection of the three sublattices x_1x_2, x_2x_3, $x_1x_4x_5$ is a sublattice $x_1x_2x_3x_4x_5$. Under the assumption that all components are independent, the probability is $r_1r_2r_3r_4r_5$.

The inclusive-or union $L_1 \vee L_2$ of two or more sublattice events L_1 and L_2 is a lattice and might be, but is not necessarily a sublattice, because the lattice $L_1 \vee L_2$ may or may not contain both the g.l.b. and l.u.b. Since the intersection $L_1 \wedge L_2$ is double counted, the probability of the union is

$$\text{pr } L_1 \vee L_2 = \text{pr } L_1 + \text{pr } L_2 - \text{pr } L_1 \wedge L_2.$$

For example, under the independence assumption

$$\text{pr } x_1x_2 \vee x_2x_3 = \text{pr } x_1x_2 + \text{pr } x_2x_3 - \text{pr } x_1x_2x_3.$$

Under distributivity, a compound event is factored into a Boolean polynomial. For example,

$$(x_1x_2 \vee x_2x_3) \wedge x_1x_4x_5 = x_1x_2x_4x_5 \vee x_1x_2x_3x_4x_5.$$

A further reduction is possible in this case. Under the absorption rule described in Part 1, $x_1x_2x_3x_4x_5$ is a proper subset of $x_1x_2x_4x_5$. Therefore, this simplifies to just $x_1x_2x_4x_5$ with a probability of $r_1r_2r_4r_5$.

Inclusion-Exclusion: Inclusive-or Unions

The probability function for the inclusive-or union of three or more sublattice events is built up recurrently by associating sets pairwise. For three lattices L_1, L_2, L_3

$$L_1 \vee L_2 \vee L_3 = (L_1 \vee L_2) \vee L_3.$$

Therefore,

$$\begin{aligned} \text{pr } L_1 \vee L_2 \vee L_3 &= \text{pr } L_1 \vee L_2 + \text{pr } L_3 - \text{pr } (L_1 \vee L_2) \wedge L_3 \\ &= \text{pr } L_1 \vee L_2 + \text{pr } L_3 - \text{pr } (L_1 \vee L_3) \wedge (L_2 \vee L_3). \end{aligned}$$

For the three sublattice events x_1x_2, x_2x_3, $x_1x_4x_5$

$$x_1x_2 \vee x_2x_3 \vee x_1x_4x_5 = (x_1x_2 \vee x_2x_3) \vee x_1x_4x_5.$$

The probability function is

$$\begin{aligned} &\text{pr } (x_1x_2 \vee x_2x_3) \vee x_1x_4x_5 \\ &= \text{pr } x_1x_2 \vee x_2x_3 + \text{pr } x_1x_4x_5 - \text{pr } (x_1x_2 \vee x_2x_3) \wedge x_1x_4x_5 \\ &= r_1r_2 + r_2r_3 - r_1r_2r_3 + r_1r_4r_5 - r_1r_2r_4r_5. \end{aligned}$$

The generalization of this process is the method of inclusion-exclusion (IE), also known as Poincaré's theorem (cf. Feller [19, p. 99] and Riordan [49]). Let the lattice L be the inclusive-or union of m sublattice events. Let S_1 denote the sum of the m probabilities of the sublattices, S_2 the sum of the probabilities of the $\binom{m}{2}$ intersections taken two at a time, S_3 the sum of the probabilities of the $\binom{m}{3}$ intersections taken three at a time, etc., and S_m the probability of the intersection of all m sublattices. Then

$$\text{pr } L = S_1 - S_2 + S_3 - \ldots + (-1)^{m-1} S_m. \tag{1}$$

Minimal States

For system reliability analysis, L could be either the minimalized lattice of paths and its terms the "minimal path sets", or else the minimalized lattice of cuts with terms that are the "minimal cut sets". Note that the form of the function is not affected by whether or not the system has the coherence property with all common-valued components. Therefore, it applies equally well to both coherent and noncoherent systems. Since L could be either the lattice of paths or else the lattice of cuts, the designation "minimal state" is used in the sequel for the terms of a minimalized polynomial, whenever it is possible to do so.

Computer Programs

Computer software is available to generate probability functions of the form of Equation (1) and to calculate the system reliability, with the minimal states and the component probabilities as input, but only for coherent systems. Three programs of this type are SCOPE [44], MAPS [11], and SPARCS [13, 29], all of which are now available as "packages", so that the user need supply only the necessary input data.

Although all three of these programs are virtually indistinguishable in the way IE is employed, there are certain differences. For example, SCOPE is a two-pass FORTRAN program, with the function generated in the first pass and the calculation in the second pass. Both MAPS and SPARCS are one-pass PL/1 programs, with all operations in a single pass. MAPS and SPARCS also have a modularity feature so that the system can optionally be processed as a complex configuration of independent modules, with separate probability calculations for each module and for the system as a whole. Also, both SCOPE and MAPS are "estimation" programs and provide merely a single value of the reliability of the system, whereas SPARCS "assesses" the system reliability; this means that it provides a schedule of confidence levels associated with every potential value of R between zero and one. SPARCS also optionally assesses the system MTBF (mean time between failures).

SPARCS assesses by a combination of Bayesian and Monte Carlo techniques. The components are all assumed to be either attribute-type or else time-to-failure type, or any mixture of these, with no restriction as to their placement. The component reliabilities are subject to Bayesian prior distributions that are functions of the prior history data, either beta for attribute-type components or negative-log gamma for the time-to-failure case. The component values are sampled from these prior distributions and substituted into the IE system function in a series of repeated Monte Carlo trials; the result is an empirical distribution of the system reliabilities, from which the assessments are performed. A newer version of SPARCS, called SPARCS-2 [26] incorporates a "super modularity" feature. It is modular not only in the sense that the system is represented as a complex configuration of independently failing modules, each module in turn being a complex configuration of

independently failing components, but also in that modules with minimal-path reliability functions can be intermixed with minimal-cut unreliability modules, with no restrictions.

Computer Time Management: Keeping the Probability Function Small

The probability function, Equation (1), could potentially be enormous, because of the upper limit of $2^m - 1$ terms. For example, with m = 10, there could be up to 1023 terms representing the ten minimal states and all the different possible sublattice intersections of two at a time, three at a time, etc. Since the maximum increases as the power of two for additional minimal states, this could be beyond the capacities of the fastest and largest computers. Fortunately, the IE probability polynomial almost never gets this large; in most practical problems, the part that is used is a tiny fraction of the maximum. The more overlapping components there are between minimal states and the more complementation, the more terms are cancelled. Modularization and inversion can help reduce the costs even further.

Overlapping components reduce the size of the probability polynomial because most of the higher order intersections are nonexistent. Those higher order intersections generated by IE can have many multiple entries cancelling against each other, so that the ultimate size of the polynomial is a very tiny fraction of a $2^m - 1$ maximum. For example, Burris [11] reports a case of a coherent system with 33 partially overlapping components between and among 18 minimal paths, processed by the MAPS program, with a probability polynomial having only 84 terms instead of the potential maximum of $2^{18} - 1 = 262,143$.

Complementation reduces the size of polynomial by cancelling out intersections of complemented sublattices. This can be done only for noncoherent systems, because there are no complements in a minimalized lattice form with coherence. Since there is no known special software for noncoherent IE polynomial generation, there are no examples of case studies of experience with complementation.

Modularization means representing the system as a configuration of modules, generating a separate IE polynomial for each module, and another one for the system as a configuration of modules. This not only eliminates most of the irrelevant higher order intersections, but also saves processing time because fewer components and terms are processed at a time; also, the combined sum of the sizes of all IE polynomials for all of the modules is still very much less than for processing as a unit. This modularization feature is incorporated into both MAPS and SPARCS. Burris [11, p. 82] reports that for 33 components with 18 minimal paths, when the system is represented as three modules in series, there is a combined total of 20 terms in all three polynomials, all of them much shorter than the 84 terms of the probability polynomial for the integrated system; also, the processing time was reduced from 9.09 seconds to 1.5 seconds on an IBM 360/65.

"Inversion" as a form of computer-time management is related to the fact that the user has an option of doing either minimal-path reliability analysis, or else minimal-cut unreliability analysis, whichever is more convenient, has the smaller number of minimal states, requires the least computer time, etc. The minimal cuts can be obtained from the minimal paths or vice versa, by inverting the lattice polynomial, using De Morgan's theorems, and minimalizing. For a coherent system, this is particularly convenient, since the only simplifying operation needed is absorption. For a

noncoherent case, because there are complementary terms, a more complex form of minimalization is needed such as the reversal method discussed in [31, 33, 35]. In reference [32], a case is reported of a software reliability analysis problem originally due to Brown and Lipow [10] processed by SPARCS-2 with artificial data. There are 13 components. One module has nine minimal paths or 26 minimal cuts. The IE polynomial for the minimal paths has 91 terms and required 31 seconds of CPU time (370/158) for 100 simulations; the polynomial for the minimal cuts has 421 terms and required 152 seconds for 100 trials.

APPROXIMATIONS

Because the reliability R is frequently estimated for systems on the order of .95, .99, .999, or even higher, the probability of failure $\bar{R} = 1 - R$ can be very small. By approximating, we may either overestimate or underestimate $\bar{R}$ but have little noticeable effect upon the estimate of R. Approximations are based on the idea that if the probability of a failure for a single component is very small, the probability of having simultaneous failures of two or more components may be insignificant and negligible. The approximations which are employed include: serializing methods, such as "parts count" and "single point failures"--these effectively delete minimal cuts with two or more components; methods that delete unimportant components, cut sets, or higher order intersection terms from IE generated polynomials--this includes the popular "singles, doubles, etc." in the fault-tree methodology which does a combination of all of these; the Esary-Proschan bounds; and error bounds.

Error Bounds

An approximation implies a willingness to accept a reasonably small error in exchange for computational feasibility. It is useful, whenever possible, to set bounds on the amount of error. These bounds depend upon the form of the approximation. Since the analyst may be mixing several different kinds of approximations and not doing any one of them in a "pure" form, but basing his way of applying each type on arbitrary numerical criteria, there is no single error bound that would ordinarily be relevant to a mixture of approximations.

For example, consider the "singles, doubles, etc." approximation, variations of which have been widely used in "risk assessment" software for fault trees. Each software package does it a little differently, but the standard treatment appears to be as follows: first, delete all components with very low failure probabilities, then all cut sets of two or more components with a combined very low failure probability, then three or more, etc., stopping, for example, at cut sets of five or more components. Then, in generating the IE probability polynomial, delete all higher order terms beyond a specified number of intersections or below a specified probability, or possibly both. Clearly, no generalized error bound can be established for a procedure that combines this many different types of approximations.

The best known and possibly most useful error-bound model is Bonferroni's inequality (see references [19, p. 110] and [39, p. 8]). This is based on truncating the IE probability polynomial, Equation (1), by deleting higher order intersection terms beyond a certain point. Suppose that only the first r - 1 terms of the right-hand side of Equation (1) are used, viz:

$$\text{pr } L \approx S_1 - S_2 + S_3 - \ldots + (-1)^{r-2} S_{r-1}.$$

Then, the maximum error (true value minus the approximation) is smaller in absolute value than the next term S_r. Note that because the polynomial is an alternating series, if r is an even number, the probability is understated, and overstated if r is an odd number. Messinger and Shooman [38] provide some Bonferroni-type error bound calculations for examples with idealized components all having common probability values, following the general Moore-Shannon approach.

A conservative error-bound model is given in [27], based on deleting minimal states under the twin assumptions that the minimal states are independent (i.e., have no overlapping components) and all have identical probabilities. If the system has m minimal states and m_1 of these are not accounted for, the error approaches $1 - \exp\{-m_1/m\}$. Since

$$x \approx 1 - \exp\{-x\}, \text{ for } x < .10,$$

this implies, for example, that if up to ten percent of the minimal states are unknown or not used, the maximum error in the estimate of the system probability is approximately equal to the proportion of missing states.

This model overstates the error for most practical problems because the usual practice is to delete only states (i.e., cut sets) having very low probabilities and to retain those with high probabilities. Likewise, since the minimal states are in general mutually dependent (i.e., with overlapping components), the independence assumption causes a larger error than would be obtained in practice. On the other hand, this model provides useful information in the sense of the potential effect of missing data. For example, suppose a system unreliability study is performed, and the analysis inadvertently or erroneously yields a fault tree that does not account for all of the relevant system failure modes. Then, based on the possibility

that the unknown cut sets could have a reasonably high probability, the maximum error $1 - \exp\{-m_1/m\}$ does not seem to be all that conservative. For further discussion on the potential effect of missing information, refer to Barlow and Lambert [3].

Serializing: "Parts Count" and "Single Point Failure"

Serializing is a form of approximation which has a very simple model to calculate an estimate of the system reliability. The simplifying assumptions are threefold: (1) the system is logically "serial"--the only minimal cut sets are individual failures, and all components are needed for system performance; (2) the components all have extremely low (constant) failure rates with all individual failures governed by the exponential distribution; and (3) all components have the same duty cycle.

An example of serializing is "parts count" (cf. Bazovsky [5, pp. 90-91]) which is often used for complex circuits consisting of very many parts of several different types, such as, for example, microminiaturized electronic circuits with many thousands of individual elements: so many diodes of each different type, so many transistors, so many resistors, etc. Let the system consist of m different types, with $n_1, n_2, \ldots, n_m$ elements of each type, respectively, and failure rates $x_1, \ldots, x_m$. To get rid of the "nuisance" parameter "mission time", scale time in "mission equivalent" units; therefore, the failure rates x_i are "failures per mission". Under the exponential assumption, for each type i the component reliability is simply $\exp\{-x_i\}$. Because of the very high reliability assumption, the failure rate for each type is approximately equal to its unreliability, that is,

$$\exp\{-x_i\} \approx 1 - x_i, \quad i = 1, \ldots, m.$$

The system reliability is approximately

$$\exp\left\{-\sum_{i=1}^{m} n_i x_i\right\} \approx 1 - \sum_{i=1}^{m} n_i x_i,$$

and $\sum_{i=1}^{m} n_i x_i$ may be used interchangeably as the approximate system unreliability or else the approximate system failure rate.

There is clearly an error in this approximation, but we cannot tell in advance how much or in which direction without a case-by-case analysis. The method overstates R in the sense that not all cut sets are accounted for; R is also understated, however, in the sense that the approximate system unreliability $\sum_{i=1}^{m} n_i x_i$ is for all practical purposes the sum of the first order terms of Equation (1). Because all terms after the first one are deleted, $\bar{R} = 1 - R$ is overstated, and R understated.

Another serializing method similar to parts count is "single point failure (spf)" analysis. The assumptions are similar to those in parts count, and the method of calculation the same. The difference, if any, is one of degree; spf is usually applied to large systems having relatively few components, perhaps one, two, or three of each type. Although the system is complex and possibly may have many redundancies, identify, isolate, and concentrate attention upon the "single point failures", those failure modes which could individually cause failure of the system because no adequate backup is available. Because this requires detailed study to isolate the most significant potential trouble spots, spf should be viewed more as a management tool than a method of estimating the system reliability.

Approximating the System Failure Rate and R with a Fault Tree

An approximation is presented in this section which has been widely used in the fault-tree methodology. In principle, it is a form of serializing,

like "parts count". Instead of serializing on the components, however, the serializing is on the minimal cut sets. For a system with m minimal cuts, let x_i, i = 1, ..., m denote the net failure rate of cut set i, that is, the contribution to system failure rate, in failures per "mission equivalent" time unit. Since for each component the failure rate and the unreliability are approximately the same, x_i is the product of the failure rates of the components in cut set i. Accordingly, the system failure rate is approximately $\sum_{i=1}^{m} x_i$. As was shown above, if time is in "mission equivalent" units, $\sum_{i=1}^{m} x_i$ is also approximately equal to the system unreliability $\bar{R}$, and R is therefore approximately $1 - \sum_{i=1}^{m} x_i$. Since $\sum_{i=1}^{m} x_i$ is an approximation to the first term of Equation (1), it can be seen that this method overstates $\bar{R}$ and understates R. The one advantage it seems to have is speed, particularly for hand computations. For further background, refer to Fussell [21].

The Esary-Proschan Upper and Lower Bounds

In references [16] and [17], Esary and Proschan present an approximation based on the assumption that the minimal states are independent and have no elements in common, similar to those incorporated into the conservative error-bound model in reference [27] described above. The independence assumption is equivalent to parallel-series, parallel in the minimal states and series in the components within minimal states. It also overstates the probability which is being approximated. Let the system have m minimal states, either paths or else cuts, with probabilities x_i, i = 1, ..., m, respectively. The approximate probability is

$$P = 1 - \prod_{i=1}^{m} (1 - x_i).$$

For example, if the $\{x_i\}$ are minimal paths, P is greater than the system reliability R; this is called the "upper bound on the system reliability". If the $\{x_i\}$ are the minimal cuts, P is greater than the unreliability 1 - R; 1 - P is called the "lower bound on the system reliability". In reference [28], there is a discussion of the upper and lower bounds and a comparison to exact results for a five-component bridge circuit. It showed that for highly reliable components, the upper bound greatly overstates R, and is unusable. The lower bound, however, appears to yield fairly accurate results--the more reliable the components are, the greater the accuracy.

SUMMARY

This paper attempts to amalgamate the logic of system reliability analysis with the computational aspects. A system reliability function is derived by inclusion-exclusion from the minimal states, the terms of a minimalized lattice polynomial of system states. Both exact methods and approximations are covered. The theoretical bases of several widely used approximations are discussed. Related topics include coherence versus non-coherence, error bounds, computer software, and fault-tree methodology.

REFERENCES

[1] Atkinson, John H., "The Set Equation Transformation System Used in Analysis of a Typical Naval Weapon System," in [2], pp. 187-202.

[2] Barlow, Richard E., Jerry B. Fussell and Nozer D. Singpurwalla, Reliability and Fault Tree Analysis. Philadelphia: Society for Industrial and Applied Mathematics, 1975.

[3] Barlow, R. E. and H. E. Lambert, "Introduction to Fault Tree Analysis," in [2], pp. 7-35.

[4] Barlow, Richard E. and Frank Proschan, Statistical Theory of Reliability and Life Testing. New York: Holt, Rinehart and Winston, 1975.

[5] Bazovsky, Igor, Reliability Theory and Practice. Englewood Cliffs, N.J.: Prentice-Hall Inc., 1961.

[6] Bell Telephone Laboratories, "Launch Control Safety Study," Section VII, Vol. 1 (1961).

[7] Birkhoff, Garrett, Lattice Theory (3rd ed.). Providence: American Mathematical Society, 1967.

[8] Birnhaum, Z. W., J. D. Esary and S. C. Saunders, "Multi-Component Systems and Structures and Their Reliability," Technometrics, Vol. 3 (1961), pp. 55-77.

[9] Boole, George, An Investigation of the Laws of Thought (1854). New York: Dover Publications (reprinted).

[10] Brown, J. R. and M. Lipow, "Testing for Software Reliability," TRW Systems, Redondo Beach, California, TRW-SS-75-02, January, 1975.

[11] Burris, Jimmy L., "Model for the Analysis of the Probability of Systems," MBA Research Report, Department of Administrative Sciences, Oklahoma State University, 1972.

[12] Chu, B. B. and D. P. Gaver, "Stochastic Models for Repairable Redundant Systems Susceptible to Common Mode Failures," in [22], pp. 342-370.

[13] Cooley, John W., "Simulation Program for Assessing the Reliability of Complex Systems," Ph.D. Dissertation, Oklahoma State University, 1976.

[14] Eagle, Kenneth H., "Fault Tree and Reliability Analysis Comparison," Proceedings 1970 Annual Symposium on Reliability, pp. 12-17.

[15] Epler, E. L., "Diversity and Periodic Testing in Defense Against Common Mode Failure," in [22], pp. 269-288.

[16] Esary, James D. and Frank Proschan, "The Reliability of Coherent Systems," in Wilcox, Richard H. and William C. Mann (eds), Redundancy Techniques for Computing Systems. Washington: Spartan Books, 1962.

[17] Esary, James D. and Frank Proschan, "Coherent Structures of Nonidentical Components," Technometrics, Vol. 5 (1963), pp. 191-209.

[18] Esary, J. D. and H. Ziehms, "Reliability Analysis of Phased Missions," in [2], pp. 213-236.

[19] Feller, William, An Introduction to Probability Theory and Its Applications, Vol. I, 3rd edition. New York: John Wiley and Sons, Inc., 1968.

[20] Fréchet, Maurice, Les probabilités associées a un système d'événements compatibles et dépendants (in two parts). Paris: Actualites Scientifiques et Industrielles, Number 859 (1940); Number 942 (1943).

[21] Fussell, Jerry B., "How to Hand-Calculate System Reliability and Safety Characteristics," IEEE Transactions on Reliability, Vol. R-24 (1975), pp. 169-174.

[22] Fussell, Jerry B. and G. R. Burdick (eds.), Nuclear Systems Reliability Engineering and Risk Assessment, Philadelphia: Society for Industrial and Applied Mathematics, 1977.

[23] Haasl, D. F., "Advanced Concepts in Fault Tree Analysis," Proceedings System Safety Symposium, Seattle, 1965, University of Washington College of Engineering and Boeing Airplane Co.

[24] Halmos, Paul R., Lectures on Boolean Algebras, Princeton, N.J.: D. Van Nostrand Company, Inc., 1963.

[25] Lambert, H. E., "Measures of Importance of Events and Cut Sets in Fault Trees," in [2], pp. 77-100.

[26] Lee, Keun Kuk, "Rejection Methods for Generating Random Deviates and Their Application in System Reliability," MS Thesis, Oklahoma State University, 1977.

[27] Locks, Mitchell O., "The Maximum Error in System Reliability Calculations by Using a Subset of the Minimal States," IEEE Transactions on Reliability, Vol. R-20 (1971), pp. 231-234.

[28] _____, "Exact Minimal-State System Reliability Analysis," Proceedings: Computer Science and Statistics, Fifth Annual Symposium on the Interface (1971), Western Periodicals Company, 13000 Raymer Street, North Hollywood, California 91605, pp. 180-192.

[29] _____, "Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment," (1975), Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, AFFDL-TR-75-144.

[30] _____, "Monte Carlo Bayesian System Reliability- and MTBF-Confidence Assessment," 1976 Proceedings of Statistical Computing Section, American Statistical Association, 806 15th Street, N.W., Washington, D.C. 20006, pp. 201-206.

[31] _____, "Reversal in Boolean Minimalization," Logique et Analyse, Vol. 18 (1976), pp. 285-297.

[32] _____, "Bayesian Methods Applied to System Reliability-Confidence Assessment," 31st Annual Technical Conference Transactions, American Society for Quality Control, Philadelphia Marriott Hotel, May 16-18, 1977, pp. 497-505.

[33] _____, "Priority of Operations in the Reversal Method of Boolean Simplification," Oklahoma State University, College of Business Administration Working Paper 77-12, April 15, 1977.

[34] _____, "Logical and Probability Analysis of Systems," Notre Dame Journal of Formal Logic, Vol. 19 (1978), pp. 123-126.

[35] _____, "Minimalization of Boolean Polynomials, Truth Functions and Lattices," Notre Dame Journal of Formal Logic, Vol. 19 (1978), pp. 264-270.

[36] Mazumdar, M., "Importance Sampling in Reliability Estimation," in [2], pp. 153-163.

[37] Mearns, A. B., "Fault Tree Analysis: The Study of Unlikely Events in Complex Systems," System Safety Symposium, Seattle, 1965.

[38] Messinger, Martin and Martin L. Shooman, "Reliability Approximations for Complex Structures," 1967 Annual Symposium on Reliability Proceedings, Washington, D.C., pp. 292-301.

[39] Miller, Rupert G., Jr., Simultaneous Statistical Inference. New York: McGraw-Hill Book Company, 1966.

[40] Mine, Hisashi, "Reliability of Physical System," Institute of Radio Engineers, Transactions on Circuit Theory Special Supplement, Vol. CT-6 (May, 1959), pp. 138-151.

[41] Moore, E. F. and C. E. Shannon, "Reliable Circuits Using Less Reliable Relays," Journal of the Franklin Institute, Vol. 262 (1956), pp. 191-208, Part II, pp. 281-297.

[42] Nagel, P. M., "Importance Sampling in Systems Simulation," Annals of Reliability and Maintainability, Vol. 5 (1966), pp. 330-337.

[43] Nagel, P. M. and R. J. Schroder, "The Efficient Simulation of Rare Events in Complex Systems," The Boeing Airplane Company, Seattle, 1967.

[44] NASA's COSMIC (The University of Georgia, Barrow Hall, Athens, GA 30601), "Exact Minimal Path and Minimal-Cut Techniques for Determining System Reliability," Program MFS-16499; "System for Computing Operational Probability Equations (SCOPE)," Version I (for IBM 7094), Program MFS-16410, Version II (for IBM 360), Program MFS-24035; "Apportionment-Prediction (APRDCT)," Program MFS-24034.

[45] Powers, Gary J., Frederick C. Tompkins and Steven A. Lapp, "A Safety Simulation Language for Chemical Processes: A Procedure for Fault Tree Synthesis," in [2], pp. 57-75.

[46] Quine, W. V., "The Problem of Simplifying Truth Functions," American Mathematical Monthly (1952), Vol. 59, pp. 521-531.

[47] _____, "A Way to Simplify Truth Functions," American Mathematical Monthly (1955), Vol. 62, pp. 627-631.

[48] _____, "On Cores and Implicants of Truth Functions," American Mathematical Monthly (1959), Vol. 66, pp. 755-760.

[49] Riordan, John, An Introduction to Combinatorial Analysis. New York: John Wiley and Sons, Inc., 1958.

[50] Rutherford, D. E., Introduction to Lattice Theory. New York: Hafner Publishing Company, 1965.

[51] Salem, S. L., G. E. Apostolakis and D. Okrent, "A Computer-Oriented Approach to Fault-Tree Construction," Electric Power Research Institute, 3412 Hillview Avenue, Palo Alto, California 94304, EPRI NP-288 (November 1976).

[52] Saunders, Sam C., "Birnbaum's Contributions to Reliability Theory," in [2], pp. xv-xxxix.

[53] Schroder, R. J., "Fault Trees for Reliability Analysis," Proceedings 1970 Annual Symposium on Reliability, pp. 198-205.

[54] Shannon, Claude, "A Symbolic Analysis of Relay and Switching Circuits," Transactions of the American Institute of Electrical Engineers, Vol. 57 (1938), pp. 713-723.

[55] U.S. Nuclear Regulatory Commission, "Reactor Safety Study, An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants," WASH-1400 (NUREG 75/014), October, 1975.

[56] Vesely, W. E., "Analysis of Fault Trees by Kinetic Tree Theory," Idaho Nuclear Corporation for U.S. Atomic Energy Commission, IN-1330, October, 1969.

[57] _____, "A Time-Dependent Methodology for Fault Tree Evaluation," Nuclear Engineering and Design, Vol. 13 (1970), pp. 337-360.

[58] _____, "Reliability Quantification Techniques Used in the Rasmussen Study," in [2], pp. 775-803.

[59] von Neumann, John, "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," Collected Works, Vol. 5, New York: The Macmillan Company (1963), pp. 329-378.

[60] Wagner, D. P., C. L. Cate and J. B. Fussell, "Common Cause Failure Analysis Methodology for Complex Systems," in [22], pp. 289-313.

[61] Worrell, Richard B., "Using the Set Equation Transformation System in Fault Tree Analysis," in [2], pp. 165-183.

[62] Worrell, Richard B. and G. R. Burdick, "Qualitative Analysis in Reliability and Safety Studies," IEEE Transactions on Reliability, Vol. R-25 (1976), pp. 164-170.

DIVISION 2

MONTE CARLO SAMPLING FROM BETA AND GAMMA PRIOR DISTRIBUTIONS IN THE "SPARCS" SYSTEM RELIABILITY- AND MTBF-CONFIDENCE ASSESSMENT PROGRAM

by

Mitchell O. Locks
Professor of Management Science
Oklahoma State University

Keun K. Lee
Program Analyst
American Airlines
Tulsa, Oklahoma

ABSTRACT*

The SPARCS-2 program for system reliability- and MTBF-confidence assessment employs Monte Carlo sampling from Bayesian conjugate prior distributions (c.p.d.'s) on the component reliabilities. Deviates are generated for two different families, "beta" for zero-one attributes (success or failure, pass-fail, go-no go, etc.) subject to a Bernoulli process, and "negative-log gamma" for constant failure-rate times to failure subject to a Poisson-exponential process. In both cases, new "rejection" methods are used which are at least as accurate as and more efficient than older methods.

The beta generator, an original one designed by Professor J. Chandler, employs a modified improper Cauchy function as an auxiliary distribution. The gamma generator was originally designed by Marsaglia [2], and it is also incorporated into the McGill Random Number Package. A negative-log gamma deviate is obtained from the gamma by a change of variables, since there is a one-to-one relationship between the two distributions. Both generators were programmed in PL/1 by K. K. Lee. The beta generator is named PROCEDURE CABTA and the gamma generator is PROCEDURE RGAMA. The purpose of this report is to describe these procedures, including a discussion of the theory of rejection methods.

*This report is based upon the junior author's MS thesis in Computer Science [3]. The assistance of Professor J. Chandler of Oklahoma State University in preparing it is gratefully acknowledged, however the authors assume all responsibility for errors.

TECHNICAL DISCUSSION: INVERSION AND REJECTION METHODS

Inversion

There are two principal classes of Monte Carlo sampling methods from continuous distributions, "inversion" and "rejection." The difference is that inversion inverts only the distribution function, whereas rejection is a more complex eclectic technique involving the probability density function (p.d.f.). Rejection methods also incorporate two-step Monte Carlo and frequently include inversion. For a random variable X with d.f. $F(x) = pr(X \leq x)$ and p.d.f. $f(x) = \frac{dF(x)}{dx}$, inversion may generally be described as follows: Generate a uniformly distributed pseudorandom variate u, $0 \leq u \leq 1$, from the uniform distribution U(0,1), representing the d.f. F(x). Then invert to obtain the corresponding percentage points $x = F^{-1}(u)$, by formula, if it is possible to do so, or by an approximating polynomial or by some iterative method if necessary.

Familiar examples of inversion are the uniform distribution U(0,1), the one-parameter exponential distribution with parameter λ, and the N(0,1) normal distribution with zero mean and unit variance. For the uniform distribution, the process is trivial; the pseudorandom variate $u = F(u)$ is the result desired. This is usually obtained by a congruential generator. For the exponential distribution, the d.f. is

$$F(x) = 1 - \exp\{-\lambda x\}.$$

With u substituted for F, the corresponding exponential random variate is

$$x = -\frac{\ell n(1 - u)}{\lambda}.$$

This is an example of inversion by formula. The N(0,1) normal distribution

is an example of inversion by an approximating polynomial, since the Hastings approximations [1, p. 191, p. 192] are usually used for this purpose.

Other examples of distributions which can be inverted easily are the Weibull, and the Cauchy and other one-parameter distributions. The Weibull can be standardized easily, and the one-parameter distributions are inverted by formula. Aside from these examples, however, there are relatively few cases where inversion is convenient or simple. This is due to the fact that most families have two parameters or more and cannot readily be standardized. Iterative inversion, which is employed sometimes, can be a very clumsy and time-consuming process because of the difficulties of converging within a reasonable amount of computer time.

Rejection Methods

There is a variety of two-step rejection generators based on the p.d.f. For each deviate, the first step is a trial value. This trial value is either "accepted" or "rejected" in the second step, depending on a second pseudorandom number, so that the values accepted are in proportion to the p.d.f. height. In the sequel, we discuss a form of "geometric" rejection with an approximating or auxiliary distribution at the first step to generate a trial deviate. This type of rejection, which is exact in the sense that it depends upon an exact relationship between the auxiliary distribution and the parent being generated, is incorporated into SPARCS in both the beta generator CABTA and the gamma generator RGAMA.

In a simple "geometric" rejection scheme, the parent p.d.f. $f(x)$ is "blanketed" from above by an auxiliary function $g(x)$, $g(x) > f(x)$,

where g(x) is such that trial random deviates can easily be generated from a p.d.f. equal to

$$\frac{g(x)}{\int_{-\infty}^{\infty} g(x)dx} .$$

In principle, g is a "little taller" and a "little fatter" than f, so that the two functions do not intersect, except possibly at a logical point of contact, such as the mode of f. G cannot be a "proper" distribution; because g is always above the p.d.f. for the proper distribution f

$$G(\infty) = \int_{-\infty}^{\infty} g(x)dx > 1.$$

Parenthetically, it should also be noted that one way of obtaining a trial deviate is to invert G, that is

$$x = G^{-1} G(x)$$

is the trial deviate. This is the technique incorporated into both the CABTA and RGAMA generators in SPARCS.

The procedure is as follows. Obtain a trial value of the random deviate x from an operation on G. The decision as to whether to use x is based on a pseudorandom number u_2. If $u_2 \leq \frac{f(x)}{g(x)}$, "accept" x; otherwise "reject" it and repeat the process again.

A simple example of rejection is with a uniform distribution U(0,1) as the auxiliary function for a beta distribution. Let f(x), $0 \leq x \leq 1$, be the p.d.f. for a beta distribution with mode x_o and density $f(x_o)$ at the mode. Let us also designate $f(x_o)$ as a "scale factor" representing the "height" of the uniform distribution. Since the auxiliary function

is uniform, generate a first pseudorandom number $u_1 = x$ as a trial value, and calculate $f(x)$. Generate a second number u_2. If the ratio

$$\frac{f(x)}{f(x_o)} \geq u_2,$$

x is "accepted" as a random deviate under the beta distribution, otherwise it is "rejected" and another random number is drawn. The process is repeated as often as necessary to obtain an acceptable deviate x, and then again as many times as necessary for the specified number of Monte Carlo trials.

The "efficiency" of the method described in the previous paragraph, the ratio of the number of deviates used to those generated, is potentially very low for highly concentrated beta distributions having large parameters and narrow, tall "spikes", because a large number of deviates would have to be generated to obtain acceptable values. The efficiency is also surprisingly high for diffuse distributions such as those with small parameters. For example, if f were uniform, the efficiency would be 100% since all deviates would be accepted. Since rejection tends to have low efficiency and computer time is a major consideration, higher efficiency is attained if the auxiliary function g and the p.d.f. f are reasonably close to one another.

The rejection generators, PROCEDURES CABTA and RGAMA, incorporated into SPARCS are both relatively efficient because the g(x) is relatively close to f(x). The beta generator has a shifted and scaled improper Cauchy function, with a mode identical with that of the beta distribution, and obtains efficiencies of approximately 80% in most cases. The gamma generator, PROCEDURE RGAMA, employs a "squeeze-down" between two auxiliary

functions, a normal from above the gamma and an exponential from below, and has efficiencies that are commonly 90% or better.

THE BETA GENERATOR: PROCEDURE CABTA

The beta generator, PROCEDURE CABTA, employs an improper Cauchy auxiliary function. The modifications are:

1. The Cauchy mode is shifted (from zero) to coincide with the beta mode p_o.
2. The Cauchy height $g(p_o)$ at the mode is (1.1) times the beta density $f(p_o)$.
3. The beta normalized second derivative, a measure of "curvature", $\frac{f''(p_o)}{f(p_o)}$ at p_o is $(1.1)^2$ times $\frac{g''(p_o)}{g(p_o)}$.

These modifications help insure that $g(p) > f(p)$ in the vicinity of the center of the distribution, for all but some "worst cases" which are not relevant to the applications of the SPARCS program. The general procedure for deriving the Cauchy function as it is described below does not work if the beta p.d.f. has a mode at either $p = 0$ or $p = 1$. Fortunately, a simpler alternative is available in those cases, with inversion and direct integration.

The Beta p.d.f. and Mode

The beta p.d.f. is

$$f(p) = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1 - p)^{b-1}, \quad a \geq 1, \; b \geq 1, \; 0 \leq p \leq 1. \quad (1)$$

In the SPARCS application, this is a prior p.d.f. on the component reliability p, where the parameters a and b are "sufficient statistics" for the

prior test data: a ≡ number of successes + 1; and b ≡ number of failures + 1. The specifications $a \geq 1$, $b \geq 1$ insure that the distribution is unimodal, but (uniform) U(0,1) only in case there are "no data". The case "no failures (b = 1)" results in a mode at p = 1, with f(1) = a. By differentiating Equation (1) and setting the derivative equal to zero, the mode can be obtained; that is

$$p_o = \frac{a - 1}{a + b - 2} . \qquad (2)$$

The Improper Cauchy Function g(p): The General Procedure

The general procedure for the Cauchy function described below differs slightly from that already incorporated into the SPARCS CABTA, but it is mathematically simpler and obtains similar results. The major difference is the fact that in the program, the second derivatives f"(p) and g"(p) of the beta and improper Cauchy densities respectively are the measures of curvature, whereas in the description, these are normalized by dividing by the densities. Hence $\frac{f''(p)}{f(p)}$ and $\frac{g''(p)}{g(p)}$ are used instead in the description below.

The auxiliary Cauchy function has the density

$$g(p) = c \frac{s}{s^2 + (p - p_o)^2} , \; c > 0, \; s > 0, \; 0 \leq p, \; p_o \leq 1. \qquad (3)$$

The parameters are a vertical scale factor c, a horizontal scale factor s, and p_o, the mode of the beta distribution. First, find s by the relationship $\frac{f''(p_o)}{f(p_o)} = (1.1)^2 \frac{g''(p_o)}{g(p_o)}$, and then c by the relationship $g(p_o) = 1.1\, f(p_o)$. The formulas for s and c, which are given below, are very simple. The mode p_o is obtained by Equation (2).

Under the beta p.d.f. (1), the "curvature" at the mode p_o is

$$\frac{f''(p_o)}{f(p_o)} = -\frac{(a + b - 2)^3}{(a - 1)(b - 1)} .$$

Under the Cauchy density (3), the curvature at p_o is

$$\frac{g''(p_o)}{g(p_o)} = -\frac{2}{s} .$$

Since $\frac{f''(p_o)}{f(p_o)} = (1.1)^2 \frac{g''(p_o)}{g(p_o)}$, the formula for the horizontal scale factor s is

$$s = 2.42 \frac{(a - 1)(b - 1)}{(a + b - 2)^3} .$$

To obtain the vertical scale factor c, we have

$$g(p_o) = \frac{c}{s} = 1.1\ f(p_o);$$

therefore

$$c = (1.1)s\ f(p_o).$$

In CABTA, the first step in obtaining a beta deviate is inverting a pseudorandom U(0,1) number under the Cauchy function. We do this with a "proper" Cauchy distribution having the same horizontal scale factor s and location parameter p_o, but with the vertical scale factor $c = \frac{1}{\pi}$. Under these conditions the trial value generated by the first pseudorandom number u_1, $0 \leq u_1 \leq 1$, is

$$p = p_o + s \tan^{-1}[\pi(u_1 - \tfrac{1}{2})].$$

The second step is an accept-reject decision for the trial value p. Generate a second pseudorandom number u_2. If $u_2 \leq \frac{f(p)}{g(p)}$, "accept" p as a beta deviate under the distribution being inverted; otherwise "reject" p.

Special Case: b = 1

The case "b = 1" represents the Bayesian prior distribution on p when there are "no failures" in the prior component data. In this case, the beta density f(p) simplifies because (1) becomes a one-parameter distribution

$$f(p) = a\, p^{a-1}.$$

By integration, we have

$$F(p) = \int_0^p f(x)dx$$

$$= p^a.$$

Therefore, a value of the variate p can be obtained by direct inversion of a pseudorandom number u, without resorting to rejection

$$p = u^{1/a}.$$

Limitations of CABTA

Reference was made above to a few cases, not occurring in normal SPARCS problems, in which CABTA might fail. These are cases where the quantity $\frac{a-1}{b-1}$ is either very large or very small. It should be pointed out that any such failure causes g(x) to lie below f(x) at some point, and each time a beta deviate is generated, this possibility is tested one or more times. If it occurs, a message is printed and the run is terminated. Thus each

usage of CABTA constitutes a stochastic test of its validity for a particular pair of values (a,b). It has never failed on any SPARCS problem to date.

The possibility of failure for some (a,b) values occurs because the beta p.d.f. may become sharply pointed (have a large curvature) at the mode, without becoming narrow at, say, its half-maximum. An example of this is a = 2, b = 1.00001, where the p.d.f. is almost a right triangle with vertices at (0,0), (1,0), and (1,2). (The case a = 2, b = 1 gives exactly a triangular p.d.f., but this is handled separately with no possibility of failure.)

It is believed that CABTA can be modified to eliminate any possibility of failure. This may be done in the future.

The Marsaglia "Squeeze" Method of Generating Gamma Variates

PROCEDURE RGAMA incorporated in SPARCS is based upon a rejection technique called the "squeeze method" designed by Marsaglia [2]. Since that paper is being published, we reproduce herein only the relevant paragraphs from Marsaglia's paper describing the procedure.

Refer to Figure 1, which shows, for several values of the parameter a, three functions $h(x) \le g(x) \le f(x)$. The top function, f, is the normal density. The method is to choose points (X,Y) with a uniform distribution under f(x) until we get one that also lies under g(x), then exit with $W = a(sX+1-s^2)^3$, where $s = a^{-\frac{1}{2}}/3$. We may avoid testing under g most of the time by first testing under h. The functions f and h are chosen to be close to g and convenient to handle. This is the essence of the squeeze method. (f is not very close to g when $a < 1$, but the procedure is still reasonable for $\frac{1}{3} < a < 1$ because h is close to g.

ALGORITHM FOR GENERATING A GAMMA VARIATE W, DENSITY $w^{a-1}e^{-w}/\Gamma(a)$, $w > 0$, FOR ANY VALUE OF THE PARAMETER $a > 1/3$, BUT RECOMMENDED FOR $a \ge 1$.

Step 1. *Generate a standard normal random variable X. Put $Z = sX + 1-s^2$, where $s = a^{-\frac{1}{2}}/3$. If $Z \le 0$ repeat this step.*

Step 2. *Generate a standard exponential random variable E.*

Step 3. *If*

$\frac{1}{2}X^2 - E < \frac{1}{2}x_0^2 + a(Z^3-z_0^3) + (3a-1)(t+\frac{1}{2}t^2+\frac{1}{3}t^3)$, *then exit with* $W = aZ^3$,

else if

$\frac{1}{2}X^2 - E < \frac{1}{2}x_0^2 + a(Z^3-z_0^3) + (3a-1)\log(Z/z_0)$, *then exit with* $W = aZ^3$,

else go back to Step 1.

In this algorithm, $x_0 = s - \sqrt{3}$, $z_0 = 1 - s\sqrt{3}$, $t = 1 - z_0/Z$.

REFERENCES

[1] Hastings, Cecil, Jr. Approximations for Digital Computers, Princeton University Press, 1955.

[2] Marsaglia, George. "The Squeeze Method for Generating Gamma Variates", unpublished document, Computers and Mathematics, with Applications (forthcoming).

[3] Lee, Keun Kuk. "Rejection Methods for Generating Random Deviates and Their Applications in System Reliability", MS thesis, Oklahoma State University, July, 1977.

☆U.S.Government Printing Office: 1978 — 757-080/651